REMARKS/ARGUMENTS

I. Introduction:

Claims 1, 6, and 23 are amended herein. Claims 5, 10-15, 17, and 21-22 have previously been canceled. Claims 1-4, 6-9, 16, 18-20, and 23 are currently pending.

II. Claim Rejections Under 35 U.S.C. 103:

Claims 1-4, 6-9, 16, 18-20, and 23 stand rejected under 35 U.S.C. 103 as being unpatentable over U.S. Patent No. 6,754,706 (Swildens et al.).

Applicant's invention, as set forth in claim 1, is directed to a method for providing a persistent connection between a client and a real server. The method generally includes: receiving a request originating from a first client for connection to a virtual server implemented on a local director which is in communication with two or more real servers; identifying a natural class of an IP address of the first client; and determining if the local director has received and sent out connection requests from the first client or any client having the same natural class as the first client by searching a table stored on the local director and identifying previous connections created between the local director and the real servers. If the local director has received and sent out a connection request to one of the real servers from the first client or any client having the same natural class as the first client, the same real server is selected for connection with the first client. If the local director has not received and sent out a connection request to one of the real servers from the first client or any client having the same natural class as the first client, one of the real servers is selected based on load balancing. Claim 1 has been amended to clarify that the local director is in communication with a plurality of clients and the real servers and that the plurality of clients are in communication with the real servers through the local director.

The Swildens et al. patent discloses a scalable domain name system with persistence and load balancing. Each DNS server is associated with a subset of the DNS groups in the network. If a DNS server is not authoritative for a client DNS server's group, a request received by the client DNS server is forwarded to the proper DNS server. Otherwise a persistence table is checked to see if a persistent response is required for the request. As shown in Fig. 3, Swildens et al. perform the following steps at a DNS server upon receiving a request from a client DNS server: (1) check to see if the client is part of group that the server is authoritative; (2) if the server is not authoritative, the request is forwarded to the proper server and no persistence check is performed; (3) if the server is authoritative, it is determined if a persistent response is required; and (4) if a persistent response is required, the persistent entry is sent to the requestor.

a) Swildens et al. do not show or suggest determining if a local director has received and sent out connection requests from the client or any other client having the same natural class as the client by identifying previous connections and selecting the same server for connection, if the local director has received and sent out a connection request.

Swildens et al. first uses authoritative connections to identify a group of servers associated with a group of clients. If the server receiving the request is not authoritative, the server simply forwards the request to an authoritative server. If a server receiving a request is authoritative, the server performs a conventional persistence check to see if a persistent connection is required. If a persistence connection is required, a table containing the client IP address and hostnames is checked and the specific IP address for that persistent connection is provided. This does not solve the problem addressed by applicant's invention. That is, there are

situation, where a client IP address will change, thus, using the IP address to provide a persistent connection will not work. For example, a firewall may translate the network address into one or more IP addresses managed by the firewall. Source persistence, as used by Swildens et al., will not work in situations where the user's IP address changes, such as when the user resides behind a firewall or array of firewalls that use multiple IP addresses. Applicant's invention, as set forth in claim 1, implements a sticky connection despite the presence of a firewall or other network device that may modify a client IP address, by checking to see if the local director has received and sent out connection requests from the client sending the request or any client having the same natural class as the first client.

In the Response to Arguments in the Advisory Action, the Examiner describes how Swildens et al. consult a table to determine if a persistent entry exists that ties a machine IP address and hostname to an IP address. For persistent hostnames, when a DNS request comes in from a client, the DNS server checks its persistent table to see if there is a persistency entry. If there is, the server will return the persistent IP address. The Examiner refers to col. 7, lines 28-33, which describes how two servers (A and C) are authoritative for the same group of clients and that when server A makes an entry for a server into its persistence table, it will also need to let server C know about the changes to the table. This merely describes conventional persistence connections using tables and further notes that servers in the same authoritative groups need to synchronize their persistence tables since any server in the group may receive requests from a client and each will have to provide a persistent connection. Thus, the persistent connection is only provided for a single IP address mapped to a client. When a request having a persistence connection is received, the DNS server merely returns the same IP address for subsequent requests. The server does <u>not</u> determine if connection requests have previously been received from the client or any client have the same natural class. If the client address changes, due to the firewall example discussed above, the entry will not be found in the table and persistent connection will be lost. Thus, Swildens et al. do

not show or suggest selecting the same server for connection with a client if the DNS server has previously received and sent out a connection request from the client or any client having the same natural class.

b) Swildens et al. do not show or suggest a local director in communication with a plurality of clients and two or more real servers, with the plurality of clients in communication with the real servers through the local director.

Swildens et al. do not show or suggest a local director which serves as a front end to a group of servers, as set forth in applicant's claims. In contrast to using a local director, requests in the system of Swildens are received directly at a DNS server. Upon receiving a request, the DNS server checks to see if it is authoritative for the client DNS server. If it is, the request may be processed at the server. If the DNS server is not authoritative to the client DNS server's group, the request is forwarded to the proper DNS server. Swildens et al. do not teach using a local director for receiving requests, providing persistence connections, and load balancing. Since Swildens et al. do not use a central node to receive and send out connection requests, there is a significant amount of additional rerouting of requests between routers, for requests that are first sent to a router which is not authoritative for that client. This requires that each DNS server maintain a list of authoritative clients since decisions are made at each server. Since applicant uses a local director to make server selection decisions, there is no need to store or update tables containing persistent connection information at any of the real servers.

c) Swildens et al. do not show or suggest selecting a real server based on load balancing if the local director has not received and sent out a connection request to one of the servers.

Swildens et al. only perform load balancing among authoritative servers. If the DNS server receiving the request for an IP address is authoritative, then a response is sent and no load balancing is performed. Applicant's invention, as set forth in claim 1, allows for load balancing among all servers in communication with a local director if the local director has not received and sent out a connection request to one of the real servers from the client sending the request or any client having the same natural class as the client. Applicant's invention is particularly advantageous in that if a client having a natural class for which no connection has been made requests a connection, a server can be selected based strictly on load balancing with no concern for selecting an authoritative server. Also, since connections are identified in a table stored on the local director, sticky connections can be timed out after a specified period for one or more natural classes.

Accordingly, claim 1 is submitted as patentable over Swildens et al. and the prior art of record. Claims 2-4 and claims 18-20, depending directly from claim 1, are submitted as patentable for the same reasons as claim 1.

Claim 2 is further submitted as patentable over Swildens et al., which do not show or suggest selecting the same real server for all clients having the same natural class subnet. In rejecting claim 2, the Examiner refers to col. 6, lines 46-65 of the Swildens et al. patent. This section of the patent refers to latency probes which are associated with a given group of servers, which may be selected based, for example, on IP addresses. Swildens et al. are not concerned with selecting a server for connection with a requesting client based on the natural class of the requesting client.

With regard to claim 3, Swildens et al. do not address receiving a request from a firewall.

Claim 18 is submitted as patentable over Swildens et al. because they do not show or suggest updating a table each time a connection is made between a local director and real servers with a new natural class. Swildens et al. are not concerned

with tracking connections based on class since they select servers based on whether they are authoritative for a client DNS server.

Since Swildens et al. do not select servers based on a subnet mask, claim 20 is also further submitted as patentable over Swildens et al.

Claim 6 is directed to a computer program product for providing a persistent connection between a client and a server. The product includes code that receives a request for connection to a virtual server implemented on a local director; code that identifies a natural class of an IP address of said first client; and code that determines if the local director has received and sent out connection requests from said first client or any client having the same natural class as said first client by searching a table stored on the local director and identifying previous connections created between the local director and said two or more real servers.

Claim 6 is submitted as patentable over Swildens et al. for the reasons previously discussed with respect to claim 1.

Claim 7-9, depending either directly or indirectly from claim 6, are submitted as patentable for the same reasons as claim 6.

Claims 8 and 9 are further submitted as patentable for the reasons discussed above with respect to claims 2 and 3, respectively.

Claim 23 is directed to a system for providing a persistent connection between a client and a real server and is submitted as nonobvious over Swildens et al. for the reasons previously discussed with respect to claim 1.

III. Conclusion:

For the foregoing reasons, Applicant believes that all of the pending claims are in condition for allowance and should be passed to issue. If the Examiner feels that a telephone conference would in any way expedite the prosecution of the application, please do not hesitate to call the undersigned at (408) 399-5608.

Respectfully submitted,

Cindy S. Kaplan
Reg. No. 40,043

P.O. Box 2448
Saratoga, CA 95070
Tel: 408-399-5608
Fax: 408-399-5609